Package 'pycno'

September 28, 2023

Type Package

Title Pycnophylactic Interpolation

Version 1.4

Date 2023-09-28

Author Chris Brunsdon

Maintainer Francesca Bitonti <francesca.bitonti@unict.it>

Depends R (>= 4.2)

Imports methods, sp, sf

Description Given a SpatialPolygonsDataFrame and a set of populations for each polygon, compute a population density estimate based on Tobler's pycnophylactic interpolation algorithm. The result is a SpatialGridDataFrame.
Methods are described in Tobler Waldo R. (1979) <doi:10.1080/01621459.1979.10481647>.

License GPL (>= 2)

LazyLoad yes

NeedsCompilation no

Repository CRAN

Date/Publication 2023-09-28 12:00:02 UTC

R topics documented:

---

pycno-package   *Pycnophylactic Interpolation*

Description

Given a SpatialPolygonsDataFrame and a set of populations for each polygon, compute a population density estimate based on Tobler's pycnophylactic interpolation algorithm. The result is a SpatialGridDataFrame.

Details

Package:	pycno
Type:	Package
Version:	1.4
Date:	2023-09-28
License:	GPL (>=2)
LazyLoad:	yes

For use in conjunction with the sp package, computes pycnophylactic surfaces given a SpatialPolygonsDataFrame and a population for each poplygon. A pycnophylactic surface is smooth, but populations allocated to each pixel sum up to the initial polygon counts, when summed over the polygons contained in each pixel.

Author(s)

Chris Brunsdon

Maintainer: Francesca Bitonti <francesca.bitonti@unict.it>

References

Tobler, W.R. (1979) *Smooth Pycnophylactic Interpolation for Geographical Regions*. Journal of the American Statistical Association, v74(367) pp. 519-530.

Examples

```
library(sp)
# Read in data for North Carolina as a SpatialPolygonsDataFrame
#nc.sids <- readShapeSpatial(system.file("shapes/sids.shp", package="maptools")[1],
#  IDvar="FIPSNO", proj4string=CRS("+proj=longlat +ellps=clrk66"))

nc.sids <- as(sf::st_read(system.file("shape/nc.shp", package="sf")), "Spatial")
row.names(nc.sids) <- as.character(nc.sids$FIPSNO)

# Compute the pycnophylactic surface for 1974 births as a SpatialGridDataFrame
# Note probably shouldn't really base grid cells on Lat/Long coordinates
# This example just serves to illustrate the use of the function
```

```
births74 <- pycno(nc.sids,nc.sids$BIR74,0.05, converge=1)

# Draw it
image(births74)

# Overlay North Carolina county boundaries for reference
plot(nc.sids,add=TRUE)
```

Estimation of Population from Pycnophylatic Interpolation
Estimate populations for a set of zones based on pycnophylactic interpolation

---

Description

Given a `SpatialGridDataFrame` of population estimates and a set of polygons, compute a population estimate based on Tobler's pycnophylactic interpolation algorithm for each zone. The result is a vector.

Usage

```
estimate.pycno(sgdf, spdf)
```

Arguments

sgdf   A `SpatialGridDataFrame` containing the ouput of a pycnophylatic interpolation, such as those produced by `pycno`.

spdf   A `SpatialPolygonsDataFrame`, giving the polygons for which estimates are wanted.

Details

Takes the estimate of population density for each pixel, checks which polygon each pixel is in, and aggregates them. Accuracy depends on the scale of pixels in the initial interpolation.

Value

A vector in which each each pixel set at the estimated population aggregation to each zone in `spdf`.

Note

Pycnophylatic interpolation has the property that the sum of the estimated values associated with all of the pixels in any polygon equals the supplied population for that polygon. A further property is that all pixel values are greater than or equal to zero. The method is generally used to obtain pixel-based population estimates when total populations for a set of irregular polygons (eg. counties) are known.

Author(s)

Chris Brunsdon

References

Tobler, W.R. (1979) *Smooth Pycnophylactic Interpolation for Geographical Regions*. Journal of the American Statistical Association, v74(367) pp. 519-530.

See Also

pycno

Examples

```
library(sp)
# Read in data for North Carolina as a SpatialPolygonsDataFrame
#nc.sids <- readShapeSpatial(system.file("shapes/sids.shp", package="maptools")[1],
#  IDvar="FIPSNO", proj4string=CRS("+proj=longlat +ellps=clrk66"))

nc.sids <- as(sf::st_read(system.file("shape/nc.shp", package="sf")), "Spatial")
row.names(nc.sids) <- as.character(nc.sids$FIPSNO)

# Compute the pycnophylactic surface for 1974 births as a SpatialGridDataFrame
# Note probably shouldn't really base grid cells on Lat/Long coordinates
# This example just serves to illustrate the use of the functions
births74 <- pycno(nc.sids,nc.sids$BIR74,0.05,converge=1)

# Create a new 'blocky' set of zones
#blocks <- gUnionCascaded(nc.sids,1*(coordinates(nc.sids)[,2] > 36) +
#  2*(coordinates(nc.sids)[,1] > -80))

crds <- sf::st_coordinates(sf::st_centroid(sf::st_geometry(sf::st_as_sf(nc.sids)),
 of_largest_polygon = TRUE))
block_ID <- 1*(crds[,2] > 36) + 2*(crds[,1] > -80)
temp <- sf::st_as_sf(nc.sids)
temp$block_ID <- block_ID
blocks <- as(aggregate(temp, by=list(temp$block_ID), head, n=1), "Spatial")

# Plot the blocky zones
plot(blocks)
# Aggregate data to them
estimates <- estimate.pycno(births74,blocks)
# Write the estimates on to the map
text(coordinates(blocks),as.character(estimates))
```

Pycnophylactic Interpolation

Perform Tobler's pycnophylactic algorithm.

Description

Given a `SpatialPolygonsDataFrame` and a set of populations for each polygon, compute a population density estimate based on Tobler's pycnophylactic interpolation algorithm. The result is a `SpatialGridDataFrame`.

Usage

```
pycno(x, pops, celldim, r = 0.2, converge = 3, verbose=TRUE)
```

Arguments

x	A `SpatialPolygonsDataFrame` representing a set of zones for which total populations are known.
pops	A list of populations, one for each zone represented in x.
celldim	The size of the cell sides for the output SpatialGridDataFrame, or a SpatialGrid or SpatialGridDataFrame with the grid topology of the desired output.
r	A relaxation parameter for the iterative step in the pycnophylactic algorithm. Prevents over-compensation in the smoothing step. In practice the default value works well.
converge	A convergence parameter, informing the decision on when iterative improvements on the smooth surface have converged sufficiently - see details.
verbose	If set to TRUE (the default value) makes the function report the maximum change in any pixel value for each iterative step. If set to FALSE the function is silent.

Details

This method uses an iterative approach, and for each iteration notes the maximum change in a pixel. When this value falls below a certain level (actually $10^{\wedge}(-converge)$ times the largest initial pixel value) the iterations stop and the current surface is reported.

Value

A `SpatialGridDataFrame` with each pixel set at the estimated density computed with the Pycnophylactic algorithm, these values being stored in the data item dens. The extent of this object is set at by the bounding box of the input `SpatialPolygonsDataFrame`. Pixels outside of all of the polygons are set to NA.

Note

Pycnophylatic interpolation has the property that the sum of the estimated values associated with all of the pixels in any polygon equals the supplied population for that polygon. A further property is that all pixel values are greater than or equal to zero. The method is generally used to obtain pixel-based population estimates when total populations for a set of irregular polygons (eg. counties) are known.

Author(s)

Chris Brunsdon

References

Tobler, W.R. (1979) *Smooth Pycnophylactic Interpolation for Geographical Regions*. Journal of the American Statistical Association, v74(367) pp. 519-530.

Examples

```
library(sp)
# Read in data for North Carolina as a SpatialPolygonsDataFrame
#nc.sids <- readShapeSpatial(system.file("shapes/sids.shp", package="maptools")[1],
#   IDvar="FIPSNO", proj4string=CRS("+proj=longlat +ellps=clrk66"))

nc.sids <- as(sf::st_read(system.file("shape/nc.shp", package="sf")), "Spatial")
row.names(nc.sids) <- as.character(nc.sids$FIPSNO)

# Compute the pycnophylactic surface for 1974 births as a SpatialGridDataFrame
# Note probably shouldn't really base grid cells on Lat/Long coordinates
# This example just serves to illustrate the use of the function
# It is suggested to use a hihger value for 'converge' - this value justs speeds
# things up for the example.
births74 <- pycno(nc.sids,nc.sids$BIR74,0.05,converge=1)

# Draw it
image(births74)

# Overlay North Carolina county boundaries for reference
plot(nc.sids,add=TRUE)
```

Index